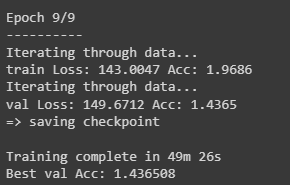
**امیرحسین احمدی 97522292 – تمرین دهم**

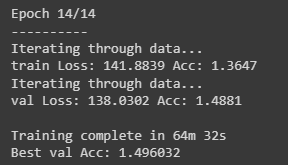
1. در ابتدا به قسمتی از تابع train که مربوط به پیداکردن best\_acc بود، اضافه کردم که همیشه وزن بهترین مدل را درون drive ذخیره کند.

الف) برای پارت اول همان نمونه داده شده را اجرا کردم با این تفاوت که pretrained را فالس کردم تا وزن ها به صورت رندوم باشند در ابتدا. همچنین مقداری data augmentation در ابتدا انجام دادم تا احتمال اورفیت کمتر شود. سپس مدل را 20 epoch آموزش دادم که با توجه به طولانی بودن این ایپوک ها همه در نوت بوک موجود نیست (همانطور که در نوتبوک دیده میشود، دو تیکه کد برای لود کردن وزن ها و اوپتیمایزر از مدل وجود دارد که در صورت استفاده نکردن از آن ها میتوان آن ها را کامنت کرد.) و فقط 10 ایپوک آخر که با هم ران کردم موجود است و نتیجه نهایی و ایپوک آخر آن را در زیر نیز میبینید.

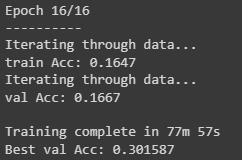
**

*با توجه به دقیق نبودن فرمول دقت در تابع ترین نتیجه ی نهایی اعتبار چندانی ندارد ولی با برسی 10 ایپوک آخر میتوان دید که دقت رفته رفته زیاد میشود و لاس کاهش میابد.*

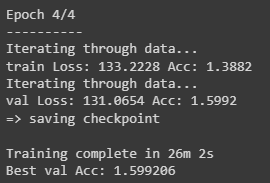
*ب) برای این قسمت ابتدا مدل pretrained را گرفته و آن را به همراه مدل کلسیفایر ساده ای که ساخته ایم به کلاس ResnetModel داده تا یک مدل کلسیفایر ساده و مدل رسنت برای feature extraction داشته باشیم. کلاس ResnetModel را نیز به گونه ای تغییر دادیم که درصورت وجود Classifier آن را به جای حالت دیفالت قرار دهد. همچنین requires\_grad پارامتر های مدل resnet را فالس کرده تا آموزش ندیده و freeze شوند. در آخر نیز بقیه پارامتر های مدل جز پارامترهای resnet را به optimizer داده تا آن ها را optimize کند و مدل را ران میکنیم. این مدل نیز 20 ایپوک ران شده که به علت محدودیت های موجود فقط 14 ایپوک آخر آن قابل مشاهده است. نتیجه:*

**

*پ) برای این قسمت به جای classifier ساده از svm استفاده کردیم تا بتوانیم با مدل قبلی مقایسه کنیم. فقط به علت اضافه شدن svc مجمور شدم بخشی از تابع ترین را تغییر دهدم و به صورت دستی svc را optimize کنم، به این صورت که خروجی resnet را گرفته و به svc داده و مقادیر خطای آن را حساب کرده و اگه در فاز train بودیم آن را با label های داده شده فیت میکنیم. من سعی کردم که تا حد امکان بقیه تابع را تغییر ندهم، ولی نتایج خیلی با نتایج مدل قبلی تفاوت دارد و همچنین میتوان گفت که اورفیت شده زیرا از یک جایی به بعد دقت ولیدیشن کاهش هم میابد. برای این سوال نیز 16 ایپوک آخر از 20 ایپوک در نوتبوک موجود است.*

**

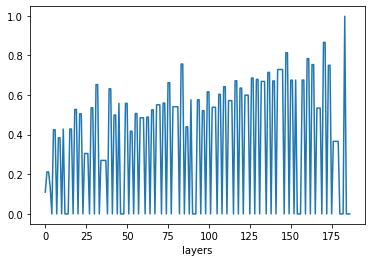
*ت) برای این قسمت، همان مدل pretrained شده قسمت ب را در نظر گرفته و این بار همه وزن ها را با هدف fine-tune آموزش میدهیم و همانطور که مشاهده میشود نتایج بعد از 5 ایپوک بسیار بهتر شده است.*

**

*ث) برای این قسمت ابتدا یک کلاس هوک (مانند callback در keras) تعریف کرده که بتوانیم خروجی هر لایه را بگیریم. آن را به انتهای همه لایه ها اضافه میکنیم. سپس برای سمپلی که برای این سوال درست کردیم خروجی را بدست آورده و به ازای هر لایه درصد مقادیر صفرش را بدست میاوریم و در نهایت پلات میکنیم. قبل از fine tune :*

**

*بعد از fine tune :*

**

*همانطور که مشاهده میشود برخی لایه ها درصد صفر بودنشان بعد از fine tune کاهش یافته است.*